IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

		§	
In re Application of:			
	Curtis R. PRIEM	§	Confirmation No.: 9182
		§	
Serial No.:	10/804,945	§	Group Art Unit: 2111
		§	
Filed:	March 19, 2004	§	Examiner: Thomas J. Cleary
		§	
For:	METHOD AND APPARATUS FOR LATENCY BASED THREAD SCHEDULING	§	

MAIL STOP APPEAL BRIEF-PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

Applicant submits this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2111 dated January 18, 2008, finally rejecting claims 1-20 (the "Final Office Action"). The final rejection of claims 1-20 is appealed. This Appeal Brief is believed to be timely since it is electronically transmitted by the due date of August 10, 2008, as set by the mailing of a Notice of Panel Decision from the Pre-Appeal Brief Review. The Commissioner is hereby authorized to charge Deposit Account No. 20-0782/NVDA/P000455/SW for any fees necessary to make this Appeal Brief timely and acceptable to the Office.

TABLE OF CONTENTS

Real Party in Interest

The present real party in interest is NVIDIA Corporation, a corporation of the State of Delaware, having a place of business at 2701 San Tomas Expressway, Santa Clara, CA 95050.

Related Appeals and Interferences

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-15 and 17, 18, 21-23 are pending in the application. Claims 1-31 were originally presented in the application. Claims 16, 19, and 20 are cancelled. Claims 1-15 and 17, 18, 21-23 stand finally rejected as discussed below. The final rejections of claims 1-15 and 17, 18, 21-23 are appealed. The pending claims are shown in the attached Claims Appendix.

Status of Amendments

All claim amendments prior to and including the Final Office Action have been entered by the Examiner.

Summary of Claimed Subject Matter

Claimed embodiments include a method of scheduling servicing of a thread (*see, e.g.*, claims 1-8, 17-18, and 21-23) and an apparatus for scheduling servicing of a thread (*see, e.g.*, claims 9-15).

Claim 1 recites scheduling threads representing requests received from hardware devices for servicing in a single queue. As claimed, the method includes the steps of (i) receiving an interrupt from a hardware device, (ii) masking interrupts from hardware devices, (iii) acquiring latency information, (iv) unmasking interrupts from the hardware devices, (v) simultaneously rearranging threads in a single queue, and (vi) ordering all requests from the hardware devices in the single queue, *Application*, paragraphs [0046]-[0047] and Figure 4. An important aspect of Appellants' invention is that the single queue represents the order in which all threads will be serviced for all of the hardware devices.

Claim 9 recites an apparatus configured to schedule threads representing requests received from hardware devices for servicing in a single queue. As claimed, the apparatus includes means for (i) receiving an interrupt from a hardware device, (ii) masking interrupts from hardware devices, (iii) acquiring latency information, (iv) unmasking interrupts from the hardware devices, (v) simultaneously rearranging threads in a single queue, and (vi) ordering all requests from the hardware devices in the single queue, *Application*, paragraphs [0046]-[0047] and Figures 2 and 4. An important aspect of Appellants' invention is that the single queue represents the order in which all threads will be serviced for all of the hardware devices.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1, 3-6, 9-10, and 23 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Zolnowsky (U.S. Patent No. 5,826,081), Browning (U.S. Patent No. 6,633,897), and Jones (U.S. Patent No. 5,812,844).

2. Claims 1-15, 17-18, and 23 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ramakrishnan (U.S. Patent No. 6,085,215), Jones, and Browning.

3. Claims 21 and 22 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Ramakrishnan, Jones, Browning, and Cheng (U.S. Publication No. 2002/0083143).

ARGUMENT

Obviousness of Claims 1, 3-6, 9-10, and 23 over Zolnowsky, Browning, and Jones

The Examiner has rejected claims 1, 3-6, 9-10, and 23 under 35 U.S.C. 103(a) as being unpatentable over *Zolnowsky* (U.S. Patent No. 5,826,081), *Browning* (U.S. Patent No. 6,633,897), and *Jones* (U.S. Patent No. 5,812,844). Appellants respectfully disagree with these rejections.

Independent claims 1 and 9 recite the limitations of (i) masking interrupts from hardware devices, (ii) unmasking interrupts from the hardware devices, (iii) simultaneously rearranging threads in a single queue, and (iv) ordering all requests from the hardware devices in the single queue. The single queue represents the order in which all threads will be serviced for all of the hardware devices. Zolnowsky, Jones, and Browning fail to teach or suggest these limitations.

In contrast to the claimed approach, Zolnowsky teaches using an array of dispatch queues instead of a single queue (see Figure 4A, col. 6, lines 30-34). Consequently, all of the requests are not ordered in a single queue, as recited in claims 1 and 9. In particular, each one of the processors disclosed in Zolnowsky has a dedicated queue, and a global dispatch (real time) queue is used for higher priority real time threads. Multiple schedulers select threads for processing from any of the queues (see Figures 5 and 7, col. 8, lines 47-54). A thread is placed in a dispatch queue and is ordered relative to the other threads in that particular dispatch queue. Because the requests are distributed among many queues, all of the requests for all of the hardware devices are not ordered in a single queue, as expressly recited in the claims. Therefore, the order in which the threads will be serviced is determined by the multiple schedulers rather than the order in which the threads are arranged any of the queues (see Figure 7, col. 8, lines 19-38).

Zolnowsky also fails to teach or suggest the limitations of masking and unmasking interrupts from hardware devices. The Examiner states that these limitations are described in col. 6, lines 34-42 of Zolnowsky, where scheduling locks are described. Zolnowsky teaches that each queue has a lock that must be acquired by a

scheduler in order to dispatch a thread from the queue. The per-queue lock is needed since any scheduler can access any queue. Nowhere does Zolnowsky suggest that any of the locks are used to mask or unmask interrupts. With regard to interrupts, Zolnowsky teaches only that "interrupt threads are always given the highest priority" (see col. 8, lines 3-4). Nowhere does Zolnowsky teach or suggest that interrupts are masked or unmasked, as recited in claims 1 and 9.

Jones also fails to teach or suggest the limitations recited in claims 1 and 9 set forth above. Jones teaches the scheduling of interrupts and other processing tasks. Jones only describes using an interrupt queue and several "lists" of threads. Further, nowhere does Jones teach or suggest ordering all requests from all of the hardware devices in the single queue. As shown in Figure 6A of the reference, a processor list, ready list, and blocked list are each maintained for scheduling purposes. Like Zolnowsky, Jones also describes using multiple queues and therefore fails to teach or suggest that all of the requests from all of the hardware devices are ordered in a single queue, as recited in claims 1 and 9. Nowhere does Jones teach or suggest that interrupts are masked or unmasked, as recited in claims 1 and 9.

Like Zolnowsky and Jones, Browning also fails to teach or suggest the limitations recited in claims 1 and 9 set forth above. In particular, Browning does not teach or suggest ordering all requests from all of the hardware devices in the single queue. In column 5, lines 8-11, Browning describes the preferred embodiment of the run queue as being "subdivided into 128 first-in-first-out (FIFO) queues, where there is a unique queue for each priority level." Rather than being a single queue, the run queue is actually made up of multiple queues, with a different queue for each priority level. The Applicant respectfully disagrees with the Examiner that the statement in col. 6, lines 1-3 of Browning, "it will be understood by those skilled in the art that various changed in form and detail may be made therein without departing from the spirit and scope of the invention" teaches or suggests the use of a single queue. A general assertion, such as this one, fundamentally cannot be used to anticipate or render obvious a specific implementation, such as that recited in the pending claims. Further, in addition to the run queues, Browning teaches using a global execution queue to list threads that are executable. A dispatcher routine is used to select a thread from the global execution

queue (col. 3, lines 27-34). Therefore, Browning fails to teach or suggest that all of the requests from all of the hardware devices are ordered in a single queue, as recited in claims 1 and 9. Browning also fails to teach or suggest that interrupts are masked or unmasked, as recited in claims 1 and 9.

Zolnowsky, Browning, and Jones fail to teach or suggest the limitations recited in claims 1 and 9. Specifically, these references fail to teach or suggest that all threads, corresponding to all requests from the hardware devices, are ordered in a single queue and that interrupts are masked or unmasked. Additionally, since none of the references can teach or suggest that all requests are ordered in a single queue, these references also fail to teach or suggest that all of the threads are simultaneously rearranged in a single queue, as recited in claims 1 and 9. Therefore, no combination of Zolnowsky, Browning, and Jones can render either claim 1 or claim 9 obvious. Since claims 2-8, 17-18, and 21-23 depend from claim 1 and claims 10-15 depend from claim 9, no combination of these references can render these claims obvious.

Obviousness of Claims 1-15, 17-18, and 23 over Ramakrishnan, Jones, and Browning

The Examiner has rejected claims 1-15, 17-18, and 23 under 35 U.S.C. 103(a) as being unpatentable over *Ramakrishnan* (U.S. Patent No. 6,085,215), *Jones*, and *Browning*. Appellants respectfully disagree with these rejections.

Ramakrishnan also fails to teach or suggest the limitations recited in claims 1 and 9 set forth above. Nowhere does Ramakrishnan teach or suggest using a queue to store processing threads. Therefore, Ramakrishnan fails teach or suggest the limitations of ordering all requests from all of the hardware devices in a single queue and that all of the threads are simultaneously rearranged in the single queue. Ramakrishnan teaches using a round-robin scheduler to select each thread for execution from a real time domain and a general purpose domain. The order in which the threads will be serviced is determined by the round-robin scheduler using real time thread flags (see Figure 2). Applicants respectfully disagree with the Examiner's assertion that col. 10, lines 48-64 of Ramakrishnan teaches that the threads are ordered according to priority, as stated in the Advisory Action. In col. 5, lines 64-67, col. 11, lines 10-16, Ramakrishnan describes that the weights are used to control the number of

consecutive processing slots that each real time thread is allocated during a round-robin scheduling cycle. Nowhere does Ramakrishnan teach or suggest that the weights are used to rearrange the order of the threads in a single queue. Further, Ramakrishnan also fails to teach or suggest that interrupts are masked or unmasked as recited in claims 1 and 9.

As previously explained, Jones and Browning fail to cure the deficiencies of Ramakrishnan relative to claims 1 and 9. Since none of the references teach or suggest the limitations recited in claims 1 and 9 that all threads, no combination of these references can render either claim 1 or claim 9 obvious. Since claims 2-8, 17-18, and 21-23 depend from claim 1 and claims 10-15 depend from claim 9, no combination of these references can render these claims obvious.

Obviousness of Claims 21 and 22 over Zolnowsky, Browning, Jones, and Cheng

The Examiner has rejected claims 21 and 22 under 35 U.S.C. 103(a) as being unpatentable over *Ramakrishnan*, *Jones*, *Browning*, and *Cheng* (U.S. Publication No. 2002/). Appellants respectfully disagree with these rejections.

Claim 21 recites the limitations that a thread is created for use during processing of a first interrupt that the one of the hardware devices is configured to generate. Claim 22 depends from claim 21 and recites the additional limitations of creating an additional thread, where a first interrupt identification number is associated with the thread, a second interrupt identification number, different than the first interrupt identification number, is associated with the additional thread, and the additional thread is created for use during processing of a second interrupt that the one of the hardware devices is configured to generate. Admitting that neither Ramakrishnan, Jones, or Browning teaches or suggest these limitations, The Examiner relies of Cheng for each of these limitations.

Cheng teaches a system for bridging between IP and non-IP networks. Nowhere does Cheng teach or suggest processing interrupts. Specifically, Cheng describes the discovery process, including the advertising of devices. However, interrupts are not used during the discovery process. In paragraph [0068], Cheng explicitly teaches the creation of threads: "one for handling device discovery, one for handling device

description, and one for handling device presentation." There is no teaching or suggestion that a thread is created for interrupt processing. Furthermore, there is no teaching or suggestion that interrupt identification numbers are associated with any of the threads.

As the foregoing illustrates, each of the four cited references fails to teach or suggest several of the limitations recited in claims 21 and 22. Thus, no combination of the cited references can render either claim 21 or claim 22 obvious.

Zolnowsky, Browning, Jones, and Ramakrishnan fail to teach or suggest the limitations recited in claims 1 and 9. Specifically, these references fail to teach or suggest that all threads, corresponding to all requests from the hardware devices, are ordered in a single queue and that interrupts are masked or unmasked. Additionally, since none of the references can teach or suggest that all requests are ordered in a single queue, these references also fail to teach or suggest that all of the threads are simultaneously rearranged in a single queue, as recited in claims 1 and 9. Therefore, no combination of these references can render these claims obvious. In view of these clear distinctions, reconsideration and allowance of all the claims is respectfully requested.

CONCLUSION

The Examiner errs in finding that:

- Claims 1, 3-6, 9-10, and 23 are unpatentable over Zolnowsky, in view of Browning, and further in view of Jones.
- Claims 1-15, 17-18, and 23 are unpatentable over Ramakrishnan, in view of Jones, and further in view of Browning.
- Claims 21 and 22 are unpatentable over Ramakrishnan, in view of Jones and Browning, and further in view of Cheng.

Respectfully submitted,

Stephanie Winner
Registration No. 52,371
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (650) 330-2310
Facsimile: (650) 330-2314
Agent for Applicant

CLAIMS APPENDIX

Claim 1 (Previously Presented): Method for scheduling the service of a thread, said method comprising the steps of:

masking interrupts from hardware devices in order to ignore interrupts for other threads;

acquiring a latency information associated with the thread, wherein the latency information indicates a time at which the thread needs to be processed;

unmasking interrupts from the hardware devices in order to detect interrupts for the other threads; and

rearranging an order in which the thread and the other threads will be serviced in a single queue to schedule the thread for processing in accordance with said latency information, wherein the rearranging is performed simultaneously for the thread and the other threads, and the thread and the other threads that are ordered in the single queue correspond to all requests received from the hardware devices.

Claim 2 (Previously Presented): The method of claim 1, wherein said latency information is computed based on a buffer size or display rate.

Claim 3 (Previously Presented): The method of claim 1, further comprising computing the time at which the thread needs to be processed by summing the latency information with a current time.

Claim 4 (Previously Presented): The method of claim 1, wherein said latency information represents a time duration that is necessary to service the thread.

Claim 5 (Previously Presented): The method of claim 1, wherein said latency information represents a maximum time allowed before a first buffer will be emptied and a read operation will switch to process a second buffer.

Claim 6 (Previously Presented): The method of claim 1, wherein said latency information represents a time duration that is necessary to setup the thread to perform interrupt processing for the thread.

Claim 7 (Previously Presented): The method of claim 1, wherein said latency information is dependent on a hardware constraint for one of the hardware devices.

Claim 8 (Previously Presented): The method of claim 1, wherein said latency information is provided by a device driver.

Claim 9 (Previously Presented): Apparatus for scheduling the service of a thread, said apparatus comprising:

means for receiving an interrupt from a hardware device;

means for masking interrupts from hardware devices in order to ignore interrupts for other threads;

means for acquiring latency information associated with the interrupt, wherein the latency information indicates a time at which the thread needs to be processed;

means for unmasking interrupts from the hardware devices in order to detect interrupts for the other threads; and

means for rearranging an order in which the thread and the other threads will be serviced in a single queue to schedule the thread to process the interrupt in accordance with said latency information, wherein the rearranging is performed simultaneously for the thread and the other threads, and the thread and the other threads that are ordered in the single queue correspond to all requests received from the hardware devices.

Claim 10 (Previously Presented): The apparatus of claim 9, further comprising computing the time at which the thread needs to be processed by summing the latency information with a current time.

Claim 11 (Previously Presented): The apparatus of claim 9, wherein said latency information is dependent on a hardware constraint for one of the hardware devices.

Claim 12 (Original): The apparatus of claim 11, wherein said hardware constraint is a size of a buffer.

Claim 13 (Original): The apparatus of claim 11, wherein said hardware constraint is a fullness of a buffer.

Claim 14 (Previously Presented): The apparatus of claim 11, wherein said hardware constraint is dynamically computed based on a buffer size or display rate.

Claim 15 (Original): The apparatus of claim 9, wherein said latency information is generated by a device driver associated with the hardware device.

Claim 16 (Canceled)

Claim 17 (Previously Presented): The method of claim 1, further comprising toggling an interrupt line.

Claim 18 (Previously Presented): The method of claim 1, further comprising:
determining the thread should be activated; and
activating the thread for processing.

Claim 19 (Canceled)

Claim 20 (Cancelled)

Claim 21 (Previously Presented): The method of claim 1, further comprising:
creating the thread prior to the steps of masking, acquiring, unmasking, and rearranging when one of the hardware devices is initialized, wherein the thread is

created for use during processing of a first interrupt that the one of the hardware devices is configured to generate; and

freeing the thread when the one of the hardware devices is shut down.

Claim 22 (Previously Presented): The method of claim 21, further comprising:

creating an additional thread, wherein a first interrupt identification number is associated with the thread and a second interrupt identification number that is different than the first interrupt identification number is associated with the additional thread and the additional thread is created for use during processing of a second interrupt that the one of the hardware devices is configured to generate; and

freeing the additional thread when the one of the hardware devices is shut down.

Claim 23. (Previously Presented): The method of claim 1, wherein the thread and at least one of the other threads correspond to interrupt requests from a single one of the hardware devices.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None